Fortify Software Security Content

2023 Update 4
December 15, 2023

About OpenText Fortify Software Security Research

The Fortify Software Security Research team translates cutting-edge research into security intelligence that powers the Fortify product portfolio – including OpenText™ Fortify Static Code Analyzer (SCA) and OpenText™ Fortify WebInspect. Today, Fortify Software Security Content supports 1,657 vulnerability categories across 33+ languages and spans more than one million individual APIs.

Fortify Software Security Research (SSR) is pleased to announce the immediate availability of updates to Fortify Secure Coding Rulepacks (English language, version 2023.4.0), Fortify WebInspect SecureBase (available via SmartUpdate), and Fortify Premium Content.

Fortify Secure Coding Rulepacks [Fortify Static Code Analyzer]

With this release, the Fortify Secure Coding Rulepacks detect 1,432 unique categories of vulnerabilities across 33+ languages and span over one million individual APIs. In summary, this release includes the following:

Improved Support for Python (version supported: 3.12)

Python is a general-purpose, powerful programming language with dynamic typing and efficient high-level data structures. It supports multiple programming paradigms, including structured, object-oriented and functional programming. This release increases our coverage to the latest version of Python expanding our support for changes in the Python standard library API. Updated existing rules coverage for the following modules:

- os
- pathlib
- tomllib

Improved Support for Django (version supported: 4.2)

Django is a web framework written in Python that is designed to facilitate secure and rapid web development. Speed and security of development are attained by the high level of abstraction in the framework, where code constructs and generation are used to drastically cut back on boilerplate code. In this release, we updated our existing Django coverage to support releases: 4.0, 4.1, and 4.2.

Improved coverage includes the following namespaces: *asyncio, django.core.cache.backends.base.BaseCache, django.db.models.Model,* and *django.middleware.security.SecurityMiddleware*. Additionally, we improved the coverage of weakness categories, which includes the following:

- Header Manipulation
- Insecure Cross-Origin Opener Policy
- Resource Injection
- Setting Manipulation

PyCryptodome and PyCrypto (version supported: 3.19.0)

PyCryptodome is a self-contained Python package that provides a comprehensive collection of cryptographic algorithms and protocols. It serves as an extended and more actively maintained version of the PyCrypto library. PyCryptodome is designed to offer a wide range of cryptographic functionalities, making it a versatile choice for developers who need to implement secure communication, data protection, and cryptographic operations in their Python applications.

Initial coverage of weakness categories includes the following:

- Key Management: Empty Encryption Key
- Key Management: Empty PBE Password
- Key Management: Hardcoded Encryption Key
- Key Management: Hardcoded HMAC Key
- Key Management: Hardcoded PBE Password
- Key Management: Unencrypted Private Key
- Password Management: Hardcoded Password
- Password Management: Lack of Key Derivation Function
- Password Management: Password in Comment
- Weak Cryptographic Hash
- Weak Cryptographic Hash: Hardcoded PBE Salt
- Weak Cryptographic Hash: Insecure PBE Iteration Count
- Weak Cryptographic Hash: Predictable Salt
- Weak Cryptographic Signature: Insufficient Key Size
- Weak Encryption
- Weak Encryption: Insecure Initialization Vector
- Weak Encryption: Insecure Mode of Operation
- Weak Encryption: Insufficient Key Size
- Weak Encryption: Stream Cipher
- Weak Encryption: User-Controlled Key Size

Detecting Risk Originating from Machine Learning (ML) and Artificial Intelligence (AI) Models

With the use of generative AI and large language models (LLMs) rapidly changing the solution space of the software industry, new risks are presenting themselves. Initial Fortify support covers Python projects that consume OpenAI API, Amazon Web Services (AWS) SageMaker, or LangChain. Support detects weaknesses resulting from implicit trust of responses from AI/ML model APIs, plus the following unique features:

Initial Support for Python OpenAI API (version supported: 1.3.8)

The OpenAI Python library enables developers to conveniently access the OpenAI REST API to interact with OpenAI models such as GPT-4 and DALL-E. The OpenAI API enables an application to send prompts to the OpenAI models and receive the generated responses as well as fine-tune existing models. The OpenAI Python module supports the ability to send and receive both asynchronous and synchronous requests powered by *httpx*. Support includes identification of potentially dangerous output from the model as well as the following new category:

- Cross-Site Scripting: AI

Initial Support for Python AWS SageMaker (Boto3) (version supported: 1.33.9)

AWS SageMaker is an offering under Amazon AWS's large umbrella of services. AWS SageMaker provides a wide set of tools to support a wide variety of ML projects from training custom models, to setting up full MLOps-supported development pipelines. Amazon's SDK for Python (Boto3) allows for

communication with a wide variety of AWS offerings, including AWS SageMaker. Support includes identification of potentially dangerous output from the model as well as the following new category:

- Cross-Site Scripting: AI

Initial Support for Python LangChain (version supported: 0.0.338)[1]

LangChain is a popular open-source orchestration framework for the development of applications using large language models (LLMs). LangChain offers tools and APIs that make it easier to create LLM-driven applications, such as chatbots and virtual agents. These are available as libraries that are based on Python and JavaScript. Support includes identification of potentially dangerous output from the model, detection of *Path Manipulation,* as well as the following new category:

- Cross-Site Scripting: AI

.NET 8 Support (version supported: 8.0.0)

As the successor of .NET 7, .NET 8 is a cross-platform, free and open-source development framework that enables programmers to write applications in different languages such as C# and VB with a standardized set of APIs. This release increases our coverage to the latest version of .NET to improve detection of weaknesses on new and existing APIs.

Expanded coverage spans the following namespaces:

- System.Collections.Frozen
- System.Net.Http.Json
- System
- System.Security
- System.Text
- System.Text.Unicode
- System.Net.Http

Java Simplified Encryption (Jasypt) (version supported: 1.9.3)

Java Simplified Encryption (Jasypt) is a small Java library used for performing password-based encryption as well as creating password digests for storage. It has integration with popular Java frameworks such as Spring, Wicket, and Hibernate.

Initial coverage of weakness categories includes the following:

- Insecure Randomness
- Key Management: Empty PBE Password
- Key Management: Hardcoded PBE Password
- Key Management: Null PBE Password
- Password Management: Lack of Key Derivation Function
- Privacy Violation: Heap Inspection

[1] LangChain is still very new. Careful consideration of security must be evaluated prior to production use.

- Setting Manipulation
- Weak Cryptographic Hash
- Weak Cryptographic Hash: Empty PBE Salt
- Weak Cryptographic Hash: Hardcoded PBE Salt
- Weak Cryptographic Hash: Insecure PBE Iteration Count
- Weak Cryptographic Hash: User-Controlled PBE Salt
- Weak Encryption
- Weak Encryption: Insecure Mode of Operation

ECMAScript 2023

ECMAScript 2023, also known as ES2023 or ES14, is the latest version of the ECMAScript standard for the JavaScript language. Key features of ES2023 include new array functions to allow changing them by copy and searching from the end. Support for ES2023 extends coverage of all relevant JavaScript weakness categories to the latest version of the ECMAScript standard.

Prototype Pollution

Prototype Pollution is a vulnerability in JavaScript applications that enables malicious users to bypass or affect business logic, along with potentially running their own code.

This Rulepack update detects if an attacker can pollute an object's prototype across the following NPM packages:

- assign-deep
- deap
- deep-extend
- defaults-deep
- dot-prop
- hoek
- lodash
- merge
- merge-deep
- merge-objects
- merge-options
- merge-recursive
- mixin-deep
- object-path
- pathval

Kubernetes Configurations

Kubernetes is an open-source container management solution for automating the deployment, scaling, and management of containerized applications. It provides container-centric infrastructure abstractions that remove dependencies on the underlying infrastructure, enabling portable deployments, and simplifies management of complex distributed systems. Improved coverage of weakness categories includes the following:

- Kubernetes Misconfiguration: Improper API Server Network Access Control
- Kubernetes Misconfiguration: Improper CronJob Access Control
- Kubernetes Misconfiguration: Improper DaemonSet Access Control
- Kubernetes Misconfiguration: Improper Deployment Access Control
- Kubernetes Misconfiguration: Improper Job Access Control
- Kubernetes Misconfiguration: Improper Pod Access Control
- Kubernetes Misconfiguration: Improper RBAC Access Control
- Kubernetes Misconfiguration: Improper ReplicaSet Access Control
- Kubernetes Misconfiguration: Improper Replication Controller Access Control
- Kubernetes Misconfiguration: Improper StatefulSet Access Control
- Kubernetes Misconfiguration: Insecure Secret Transport
- Kubernetes Misconfiguration: Insufficient Kubelet Logging
- Kubernetes Misconfiguration: Scheduler System Information Leak
- Kubernetes Misconfiguration: Uncontrolled Kubelet Resource Consumption
- Kubernetes Terraform Misconfiguration: Improper Daemon Set Access Control
- Kubernetes Terraform Misconfiguration: Improper Deployment Access Control
- Kubernetes Terraform Misconfiguration: Improper Job Access Control
- Kubernetes Terraform Misconfiguration: Improper Pod Access Control
- Kubernetes Terraform Misconfiguration: Improper Pod Network Access Control
- Kubernetes Terraform Misconfiguration: Improper RBAC Access Control
- Kubernetes Terraform Misconfiguration: Improper Replication Controller Access Control
- Kubernetes Terraform Misconfiguration: Improper Stateful Set Access Control
- Kubernetes Terraform Misconfiguration: Insecure Secret Transport

DISA STIG 5.3

To support our federal customers in the area of compliance, correlation of the Fortify Taxonomy to the Defense Information Systems Agency (DISA) Application Security and Development STIG version 5.3 has been added.

OWASP Mobile Top 10 Risks 2023

The Open Worldwide Application Security Project (OWASP) Mobile Top 10 Risks 2023 aims to raise awareness around Mobile security risks and educate those involved in Mobile application development and maintenance. To support our customers who want to mitigate Web Application risk, correlation of the Fortify Taxonomy to the initial release of the OWASP Mobile Top 10 2023 has been added.

Miscellaneous Errata

In this release, resources have been invested to ensure we can reduce the number of false positive issues, refactor for consistency, and improve the ability for customers to audit issues. Customers can also expect to see changes in reported issues related to the following:

Deprecation of Fortify Static Code Analyzer versions prior to 20.x

As mentioned in our 2023.3 release announcement, that was the last release of the Rulepacks that support Static Code Analyzer versions prior to 20.x. For this release, Static Code Analyzer

versions prior to 20.x will not load the Rulepacks. This will require either downgrading the Rulepacks or upgrading the version of Static Code Analyzer. For future releases, we will continue to support the last four major releases of Static Code Analyzer.

False Positive Reduction and Other Notable Detection Improvements

Work has continued with the effort to remove false positives in this release. Customers can expect further removal of false positives, and other notable improvements related to the following areas:

- *ASP.NET Misconfiguration: Persistent Authentication* – false positives removed in ASP.NET applications that use forms authentication services
- *Credential Management: Hardcoded API Credentials* – false positives removed from secret scanning related to HTTP Bearer tokens
- *Credential Management: Hardcoded API Credentials* – new issues detected for Avature API keys
- *Cross-Site Request Forgery* – new issues detected in NodeJS applications that use the `Express.js` JavaScript framework
- *Cross-Site Scripting* – new issues detected in Go applications that use the `html/template` package
- *Cross-Site Scripting: Reflected* – false positives removed in ASP.NET applications that use the `ListControl` class
- *Denial of Service: Format String* – incorrect mappings to OWASP Top 10 categories
- *Insecure Transport* – false positives removed in ASP.NET applications related to controller methods that handle private user data
- *Insecure Transport: Mail Transmission* – false positives removed from Python applications that use the `smtplib.SMTP` class
- *Key Management: Hardcoded Encryption Key* – false positives removed in Java applications that use the `RSAKeyGenParameterSpec` class
- *Link Injection: Missing Validation* – false positives removed in Swift and Objective-C applications that use the `WKNavigationDelegate` protocol[2]
- *Mass Assignment: Insecure Binder Configuration* – false positives removed from Java applications that use Jakarta EE APIs
- *Password Management: Password in Configuration File* – false positives removed in configuration files
- *Path Manipulation* – new issues detected in PHP applications with file uploads
- *SQL Injection* – new issues detected in NodeJS applications that use the marsdb database
- *SQL Injection: MyBatis Mapper* – new issues detected in MyBatis mapper XML files
- *String Termination Error* – false positives removed in C/C++ applications that use `printf()` and its variants
- *System Information Leak: Incomplete Servlet Error Handling* – false positives removed in Java applications
- *Weak Encryption: Insecure Initialization Vector* – false positives removed in Python applications that use the `Pycryptodome` library
- *Unreleased Resource: Streams* – false negatives identified in Java applications that use `java.nio.file` APIs
- Various dataflow false positives in Visualforce applications related to user profile information

[2] Requires Fortify Source Code Analyzer 23.1 or later

Category Name Changes

When weakness category name changes occur, merging analysis results of prior scans with new scans may result in added/removed categories.

To improve consistency, the following two categories have been renamed:

Removed Category	Added Category
Azure ARM Misconfiguration: Insecure DataBricks Storage	Azure ARM Misconfiguration: Insecure Databricks Storage
Azure ARM Misconfiguration: Insecure Redis Enterprise Transport	Azure ARM Misconfiguration: Insecure Redis Transport

Fortify SecureBase [Fortify WebInspect]

Fortify SecureBase combines checks for thousands of vulnerabilities with policies that guide users in the following updates available immediately via SmartUpdate.

Vulnerability Support

Access Control: Administrative Interface

This release includes a check to detect unsafe configuration of Spring Cloud Gateway when the gateway actuator endpoint is enabled, exposed, and not secured. In this case, attackers can create new routes and get access to internal or sensitive assets on the application's behalf. This could lead to stolen cloud metadata keys, exposure of internal applications, or denial-of-service (DoS) attacks.

Expression Language Injection: Spring

Spring Cloud Gateway versions 3.1.0, 3.0.0 to 3.0.6, and versions older than 3.0.0 contain a security vulnerability identified by CVE-2022-22947. This vulnerability allows a code injection attack when the gateway actuator endpoint is enabled, exposed, and unsecured. This release includes a check to detect if this vulnerability exists on the target server that uses the affected Spring Cloud Gateway versions.

Insecure Deployment: Unpatched Application

TeamCity On-Premises server versions 2023.05.3 and earlier are prone to an authentication bypass, which enables an unauthenticated attacker to gain remote code execution (RCE) on the server. This vulnerability has been identified by CVE-2023-42793. This release includes a check to detect this vulnerability on target servers.

Information Discovery: Undocumented API

Undocumented or limited documentation for API endpoints can provide attackers with an attack surface that is not sufficiently tested for security vulnerabilities. An attacker might perform reconnaissance to uncover deprecated, unpatched, and unmaintained endpoints to gain access to sensitive information or dangerous functionality. This release includes a check that aims to discover versioned API endpoints that are accessible but not defined in the API specification document.

Compliance Reports

DISA STIG 5.3

To support our federal customers compliance needs, this release contains a correlation of the WebInspect checks to the latest version of the Defense Information Systems Agency Application Security and Development (DISA) STIG, version 5.3.

Policy Updates

DISA STIG 5.3

A policy customized to include checks relevant to DISA STIG 5.3 has been added to the WebInspect SecureBase list of supported policies.

Miscellaneous Errata

In this release, we invested resources to further reduce the number of false positives and improve the ability for customers to audit issues. Customers can also expect to see changes in reported findings related to the following areas.

Insecure Transport: SSLv3/TLS Renegotiation Stream

TLS 1.3 does not support renegotiation. This release includes improvements for the Renegotiation Stream Injection check to reduce false positives and improve the accuracy of the results.

HTML5: Cross-Site Scripting Protection

X-XSS-Protection header is deprecated in all modern browsers. This release we have deprecated missing or misconfigured X-XSS-Protection header checks.

Fortify Premium Content

The research team builds, extends, and maintains a variety of resources outside our core security intelligence products.

DISA STIG 5.3 and OWASP Mobile Top 10 2023

To accompany the new correlations, this release also contains a new report bundle for OpenText™ Fortify Software Security Center with support for DISA STIG 5.3 and OWASP Mobile Top 10 2023, which is available for download from the Fortify Customer Support Portal under Premium Content.

Fortify Taxonomy: Software Security Errors

The Fortify Taxonomy site, which contains descriptions for newly added category support, is available at https://vulncat.fortify.com.

Contact Fortify Customer Support

OpenText Fortify
https://softwaresupport.softwaregrp.com/
+1 (800) 509-1800

Contact SSR

Alexander M. Hoole
Senior Manager, Software Security Research
OpenText Fortify
hoole@opentext.com
+1 (650) 427-9973

Peter Blay
Manager, Software Security Research
OpenText Fortify
pblay@opentext.com
+1 (669) 309-1634